Implementation of the disruption predictor APODIS in JET real time network using MARTe framework

J.M. López[1], J. Vega[2], D. Alves[3], S. Dormido-Canto[4], A. Murari[5], J.M. Ramírez[4], R. Felton[6], M. Ruiz[1],G de Arcas[1] and JET EFDA contributors*

JET-EFDA, Culham Science Centre, Abingdon, OX14 3DB, UK

[1]CAEND. Universidad Politécnica de Madrid, Spain.
[2]Asociación EURATOM CIEMAT para Fusión Madrid, Spain.
[3]Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal
[4] Dpto. Informática y Automática. UNED. Madrid, Spain.
[5] Consorzio RFX-Associazione EURATOM ENEA per la Fusione, Padova, Italy
[6] EFDA-CSU, Culham Science Center, Abingdon, OX14 3DB, UK.
** See the Appendix of F. Romanelli et al., Proceedings of the 23rd IAEA Fusion Energy Conference 2010, Daejeon, Korea .*

ABSTRACT

Disruptions in tokamaks devices are unavoidable, and they can have a significant impact on machine integrity. So it is very important have mechanisms to predict this phenomenon. Disruption prediction is a very complex task, not only because it is a multi-dimensional problem, but also because in order to be effective, it has to detect well in advance the actual disruptive event, in order to be able to use successful mitigation strategies[1]. With these constraints in mid a real-time disruption predictor has been developed to be used in JET tokamak. The predictor has been designed to run in the Multithreaded Application Real-Time executor (MARTe) framework [2]. The predictor "Advanced Predictor Of DISruptions" (APODIS) is based on Support Vector Machine (SVM) [3].

APODIS ARCHITECTURE

APODIS is based on two layers Support Vector Machine (SVM). The first version used thirteen signal [4], after a optimization process nowadays, uses seven relevant measurements: Plasma current, Mode lock amplitude, Plasma inductance, Plasma density, Diamagnetic energy time derivative, Radiated power and Total input power. These signals are processed using 32 ms time windows with a sampling frequency of 1 kHz. Various features are calculated (mean value and standard deviation of the FFT, without first component). Figure 1 shows the two layer architecture of APODIS and time diagram that visualize the 32 ms windows overlay in time.

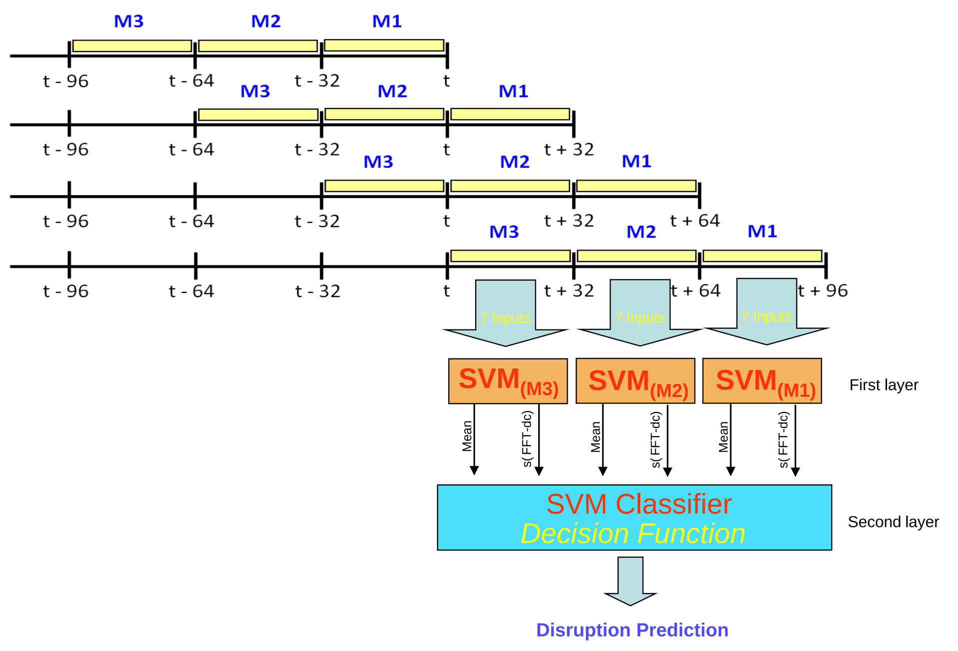

Figure 1: APODIS Architecture. First layer is formed for three RBF kernel SVM classifiers, Mx. The second layer is linear kernel classifier

APODIS ARCHITECTURE USING MARTe

MARTE is the framework to use real time applications in JET control room. MARTe provides tools for synchronisation, statistics, persistence, etc.

Figure 2 shows the APODIS architecture implemented according MARTe framework philosophy. The application has two execution threads, first collects samples for the inputs sources, these can be JET database or packets from the Real-Time Data Network (RTDN). The second one fits the data from the thread one in time packets of 32 ms. In thread one the signals can be collected from the JET data base or from RTDN network, the former is used for development cycle (test, debug, etc.) and the second one for real time working.

In thread one every signal is sampling at 1 kHz, if the sample is not available, then the last one is provide (This situation can be see like a hold effect in signals).

Later a normalization and clamping process is applied to the samples to keep signals value inside the training levels. When signals are normalize, two characteristics for the classifier are calculated (Mean, and the standard deviation of the FFT). These values and an enable flag are passed to the second thread using a Data Transfer GAM. The enable flag is set when the current plasma cross a specific threshold defined by the user, and it is used to enable the classifier operation.

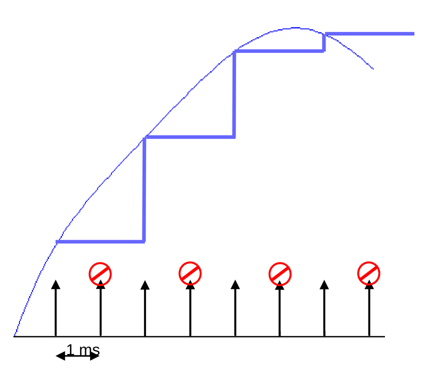

Figure 3: Hold effect in signal reading when not sample available in the RTDN.

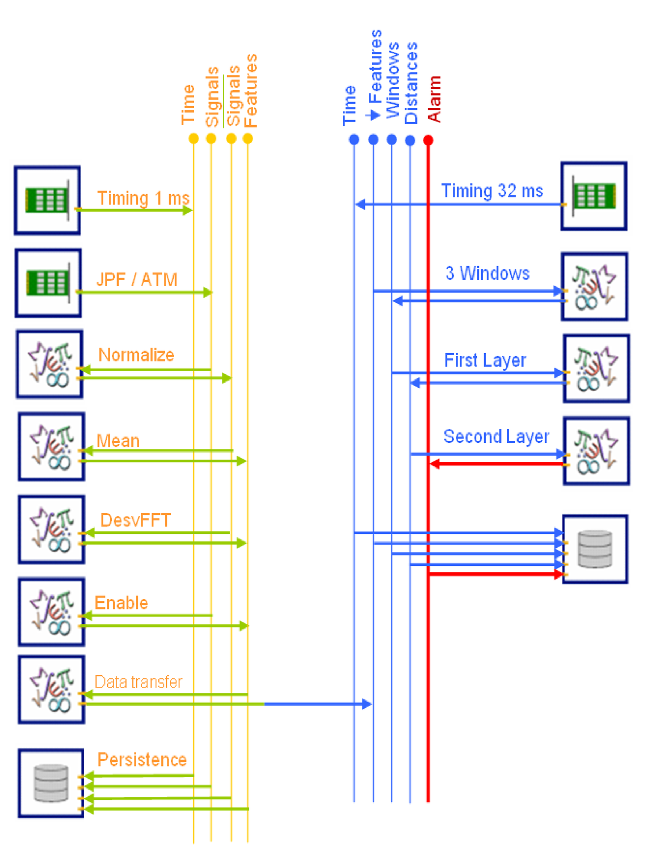

Figure 2: APODIS Architecture using MARTe.

Thread two implements the classifier operations. First a FIFO buffer is used to arrange the three windows used to feed data to the classifier, see figure 1. These data are processed for a GAM that implements the first layer of the VSM classifier. The output of the first layer is used for the second layer GAM to evaluate if the current discharge could be disruptive or not

EXPERIMENTAL IMPLEMENTATION

The real-time implementation is running in JET control room. The system is able to predict a disruption 30 ms in advance with a hit rate of 97%. It is estimated that 30 ms is a sufficient time to take protective actions. Figure 4 shows results obtained with data from C28 campaign from discharges 82429 to 82905.

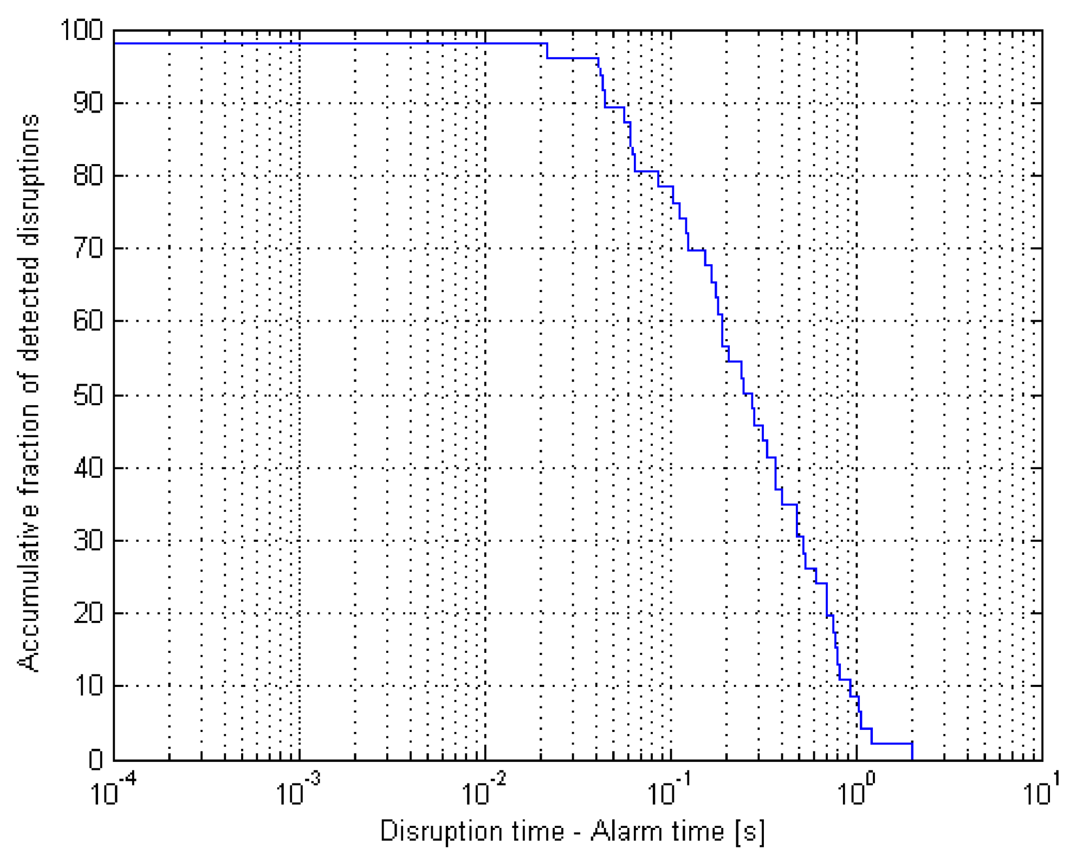

Figure 4: APODIS Accumulative prediction time for discharges 82429 to 82905 in C28 campaign.

The system has been implemented on a six core x86 architecture with an ethernet Network Interface Card (NIC) for remote administration and introspection and an Asynchronous Transfer Mode (ATM) NIC handling all real-time I/O within the JET's Real Time Data Network (RTDN). It is a user-space application running on a mainstream Linux vanilla kernel and implemented using MARTe Real-time performance has been achieved by combining available Central Processing Units (CPU) isolation and Interrupt ReQuests (IRQ) routing mechanisms.

In terms of runtime, the thread two is the most critical, in this thread the VSM is implemented. Special care has been taken in the codification of this thread to minimize execution time. The results have been very satisfactory since the average execution time of thread two is of 250 microseconds, for discharges 82429 to 82905

CONCLUSIONS and IMPLICATIONS FOR ITER

- A real-time disruption predictor has been implemented in JET.
- The predictor is accurate and reliable (High success prediction and low false alarm rates).
- The disruptions are predicted with enough time in advance (300 ms mean time).
- MARTe framework allows multithread execution, synchronization capabilities, statistic tools for performance evaluation.
- Predictor has an implementation model base in data structures. The code is unique and data structures define the software behaviour using configuration files.
- The current results are very satisfactory and the predictor is very robust. The predictor was training with data from previous campaigns with a carbon wall.
- The predictor shows a good behaviour that let thinking use it like a trigger for a mitigation system.

NEXT STEPS...

- Next step is modify the windows size to improve the resolution in prediction time. The objective will be use 1 ms windows due to the low computational time in thread two 250 micro seconds.

ACKNOWLEDGEMENTS

The authors would like to thank to Peter de Vries for the help with the JET disruption database.

This work was partially funded by the Spanish Ministry of Science and Innovation under the Project No. ENE2008-02894/FTN.

This work was supported by EURATOM and carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

REFERENCES

[1] A. H. Boozer. "Theory of tokamak disruptions" Physics of plasmas, 19, 058101 (2012) 25 pp.
[2] A. C. Neto, F. Sartori, F. Piccolo, R. Vitelli, G. de Tommasi, L. Zabeo et al. "Marte: a multiplatform real-time framework". IEEE Transactions on Nuclear Science. 57, 2 (2010) 479-486
[3] V. Cherkassky and F. Mulier. "Learning from data". (1998). New York, Wiley
[4] G. A. Rattá, J. Vega, A. Murari, G. Vagliasindi, M. F. Johnson, P. C. de Vries and JET-EFDA Contributors. "An Advanced Disruption Predictor for JET tested in a simulated Real Time Environment" Nuclear Fusion. 50 (2010) 025005 (10pp)